SPML Assignment 2 - Heuristic search for planning problems

Clemens Beissel 4547330
Christian Lammers 4578236

August 26, 2018

1 Domains

1.1 Blocks

The block domain consists of blocks. Each state of this domain can be described by 5 different predicates:

- on x y - Is block x on block y?
- ontable x - Is block x on the table?
- clear x - Is block x clear?
- handempty - Is the hand normally carrying the blocks empty?
- holding x - Is the hand holding block x?

Further does this domain have 4 different actions that can be applied:

- pick up x - Picking up a block x
- put down x - Putting down a block x
- stack x y - Stacking block x on top of y
- unstack x y - Unstacking block x of y

1.2 Logistics

The domain logistics consists of multiple objects: trucks and airplanes that form together the vehicles, package and vehicles forming physical objects and airport, location and cities forming places.

Three different predicates describe a state of this domain. A predicate describing what package is in what vehicle, a predicate describing what physical object is at what place and a third one describing what place is in what city.

The actions that can be applied in this domain are fairly straight-forward:

- loading truck
- loading airplane
- unloading truck
- unloading airplane
- drive truck
- fly airplane

2 Search algorithm and Heuristics

In order to find sufficient results three different search algorithms were used: A* search, enforced hill climbing and weighted A* search. In the following paragraphs the reasoning behind choosing these algorithms and a short explanation of them will be given as well as the choice of heuristics.

2.1 A* Search

The A* algorithm always finds a solution if it exists and this solution is always optimal if the used heuristic is admissible. Thus for problems which are not extremely oversized, the A* algorithm is an intelligent choice. The best heuristic is dependent on the problems nature and the results are therefore quiet different depending on the chosen heuristic.

2.2 Enforced hill climbing search

The enforced hill climbing algorithm always chooses nodes which value improves the value of the current node. In a problem state space this means that the algorithm chooses nodes that are higher in the state space.This is why it can get stuck in local maxima which never lead to the goal.

Even though enforced hill climbing is better than just an exhaustive search, this algorithm will encounter difficulties in terms of complexity especially if the problem search state is very large. This problem can be solved by using a heuristic.
We expect that the hill climbing algorithm will perform worse than the A* search algorithm in the blocks domain. Especially when using the blind heuristic the difference between those two algorithms will be significant.

2.3 Weighted A* search

In weighted A* search all heuristic values are multiplied by some ϵ where $\epsilon > 1$. This weighting implicates that nodes which are closer to the goal are considered earlier than nodes whose heuristic values are smaller and therefore more far away from the goal. This makes the weighted A* search more similar to greedy approach. Due to this greedy character of the algorithm the path is found much faster than the normal A* search algorithm would. On the downside might that solution not be the optimal one since less nodes had to be expanded and less actions had to be tried out.
We expect that this algorithm will find a solution much faster than the A* search algorithm. However will this algorithm also expand less nodes and the found path might not be optimal.

2.4 Heuristics

For each search four different heuristics were used:

- **blind**: basically a random heuristic; used as a baseline
- **hADD**: non-admissible heuristic that adds minimum cost of actions until goal is reached. It does not solve the relaxed task but computes an estimate of the difficulty of the problem.
- **hMAX**: admissible heuristic that takes the maximal cost of an edge in a path to the goal.
- **hFF**: Fast-forward heuristic that computes a solution for the problem and finds a path to the goal where the value of the heuristic is the number of actions in the solution. This heuristic starts at the goal state and tries to find the fastest way to the initial state by considering only actions that accommodate to the satisfaction of the goal.

3 Results

3.1 A* Search

3.1.1 Heuristics

As can be seen in figure 2 the A* search with the hADD and hFF heuristics did not need much time to find a path to the goal. This solution, however, is far from optimal because both heuristics are not admissible. In any case, they can be useful for conducting an informative search with no intention of finding the absolute best solution to the problem.

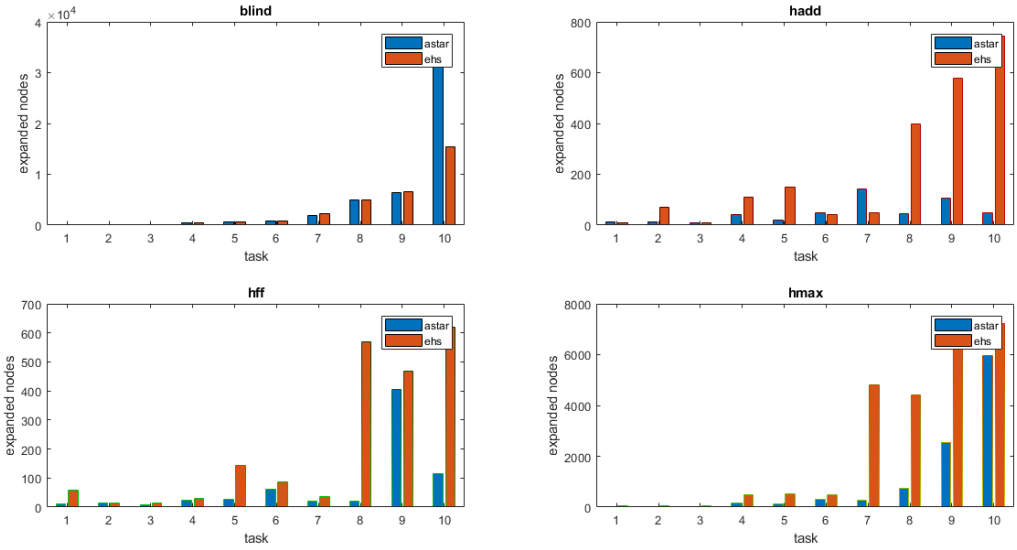

Figure 1: Complexity of the algorithms in the blocks domain with regards to the expanded nodes

Much more time was needed to find a solution by using the hMAX or the blind heuristics. When using the blind heuristic the number of expanded nodes can easily blow up especially when the state space is huge. In contrast is the space complexity of the hMAX heuristic more linear. The admissible blind heuristic can find the best path to the goal in a really short time but it can also take exceptionally long. Usually the hMAX heuristic, which is also admissible, needs more time to find the best path but its time complexity is a lot more predictable than the blind heuristic.

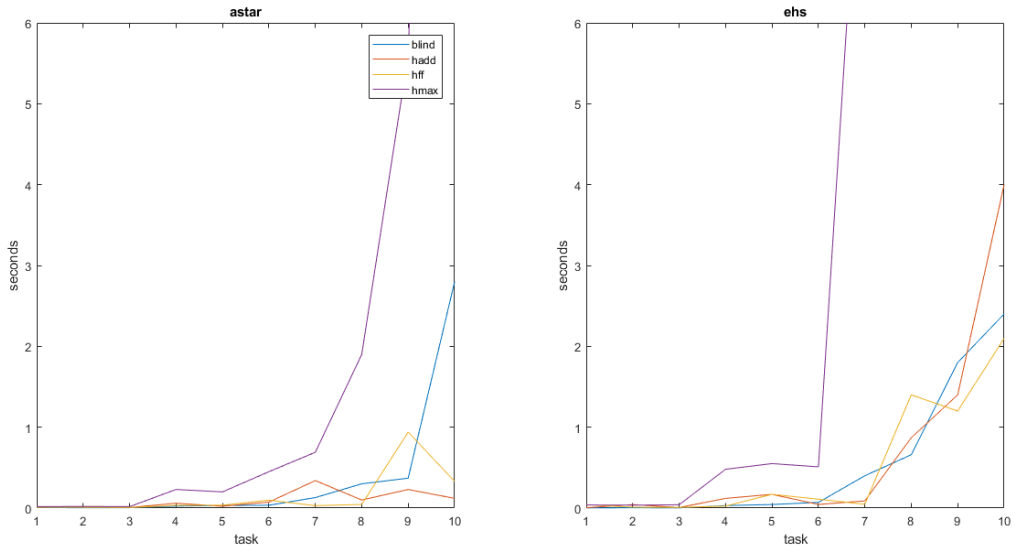

Figure 2: Complexity of the algorithms in the blocks domain with regards to the time

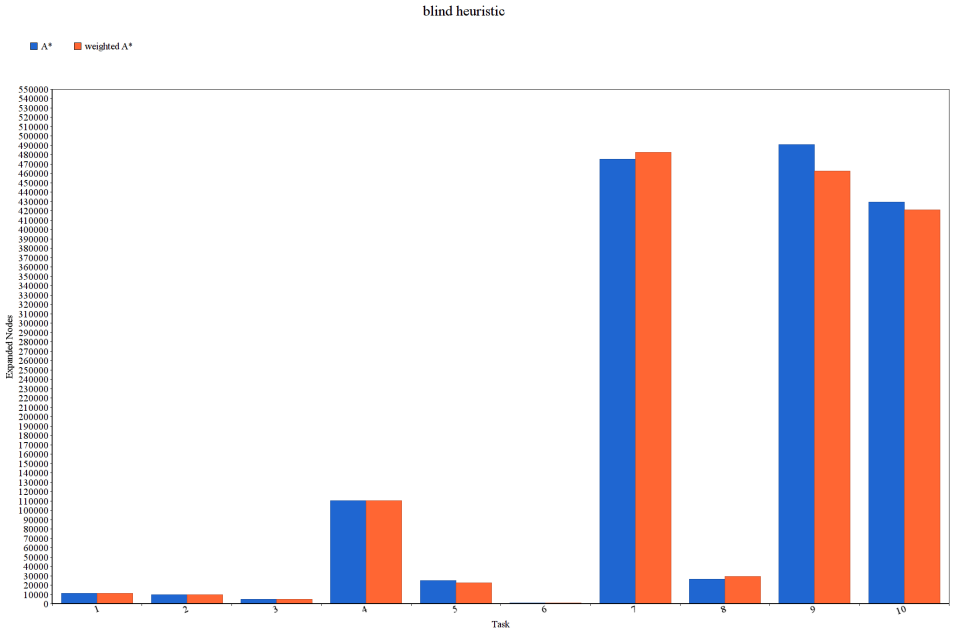

Figure 3: A* and weighted A* search in the logistics domain using blind heuristic

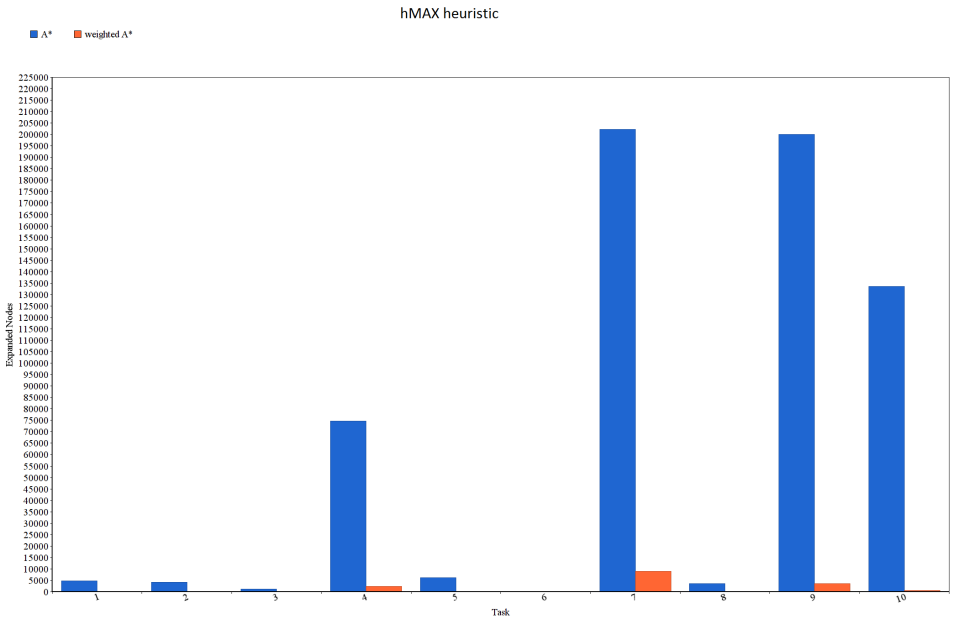

Figure 4: A* and weighted A* search in the logistics domain using hMAX heuristic

When looking at the figure 5 it becomes clear that the duration of the search is much longer with the hMAX heuristic when compared to the blind heuristic. This is always dependent on the task and the search that was used.

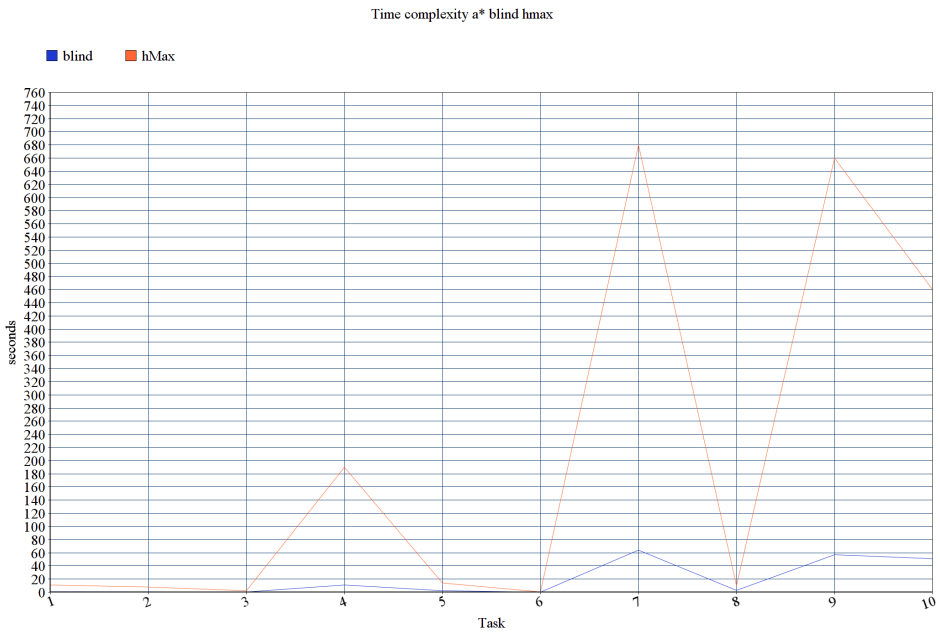

Figure 5: Time complexity of A* with blind and hMAX heuristics

Figure 6 reveals that the hFF heuristic is in general more time consuming than the hADD heuristic. Especially in the tasks 4 and 7 the time complexity is much higher compared to the hADD heuristic.

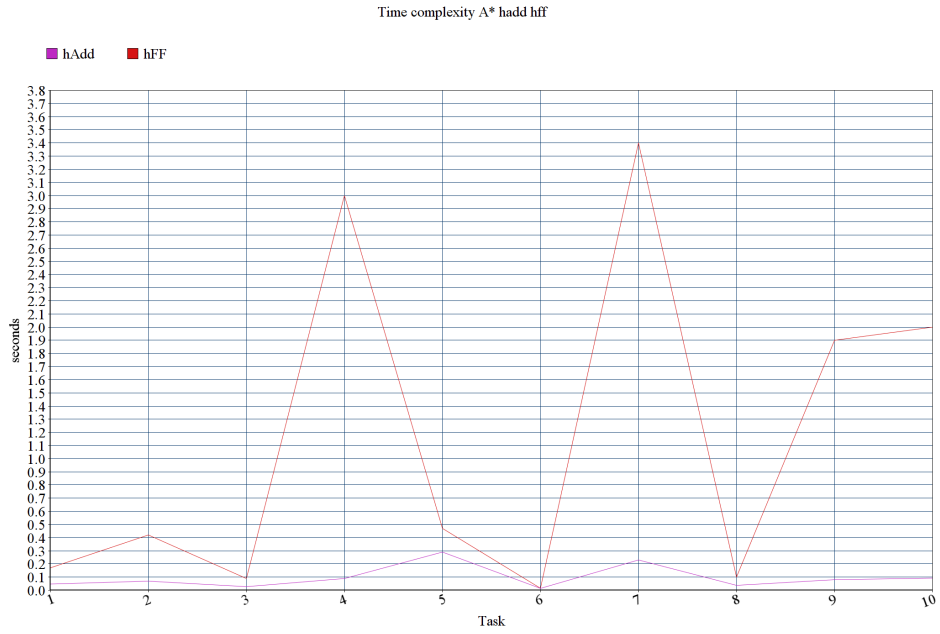

Figure 6: Time complexity of A* with hADD and hFF

3.2 Enforced hill climbing

Enforced hill climbing generally expanded more nodes than the A* search. Its time complexity was pretty identical to the A* search in the first 7 tasks but in the last three it was a bit more time consuming than the other search as can be seen in 2.

3.2.1 Heuristics

The search performed almost identical to the A* search with regards to the time complexity of the different heuristics. In figure 2 it becomes apparent that the hADD and the hFF heuristics are a little bit slower than the A* star search at around the seventh task. When looking at the space complexity, all heuristics except blind had on average a lot more expanded nodes than the A* search.

3.3 Weighted A* Search

Weighted A* Search expanded a lot less nodes than the normal A* search algorithm. Taking a look at figures 4 and 7 one can see the huge difference in expanded nodes. This is due to the weighting of the heuristic in wA* which essentially pushes the search faster to the goal while lacking in the optimality of the solution. When comparing the time complexity of the heuristics hMAX and blind one can see that in this case the blind heuristic lead the search to be completed much slower than the hMAX heuristic. This shows the unpredictability of the blind search. Regarding the other two heuristics, hADD and hFF, were more or less identical with hFF being a little bit faster than hADD.

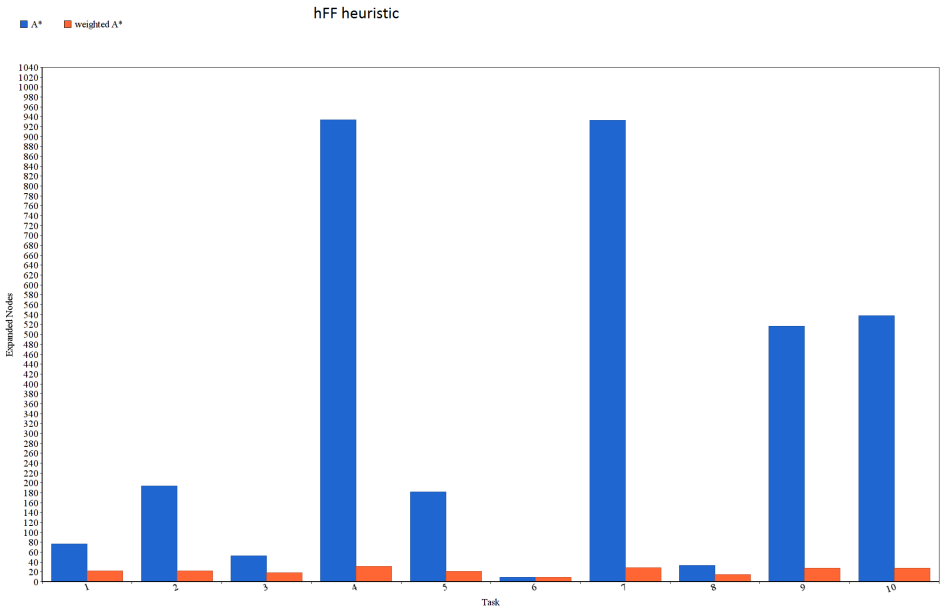

Figure 7: A* and weighted A* search in the logistics domain using hFF heuristic

3.3.1 Heuristics

The results of using the blind heuristic combined with the weighted A* search are roughly the same as when using just the blind heuristic and A* search. The reason for that is that the blind heuristic is basically no heuristic at all. Therefore no different exists between using the A* or weighted A* algorithm.

4 Conclusion

As expected, in the blocks domain the A* search performed better than the enforced hill climbing search with regards to time and space complexity. When comparing A* and weighted A* search in the logistics domain one can see that the weighted search found a solution almost instantly to every problem with surprisingly few expanded nodes. Especially looking at the tables in the appendix shows that the weighted A* algorithm found a solution much faster than the A* algorithm (see table 14 and 4).

The time it took for finding a solution was also a lot quicker when compared to the normal A* search. The most time consuming task with A* search and heuristic hMAX was task 7 with 680 seconds (table 8). The same task was completed in only 27 seconds with the same heuristic and the weighted A* search (table 15).

A Tables of the A* search algorithm

A.1 Blind heuristic

Table 1: Results of A* search with blind heuristic in the blocks domain

Task	**Expanded nodes**	**Time**	**Plan length**
1	102	0.0073	6
2	68	0.0037	10
3	65	0.0047	6
4	468	0.025	12
5	561	0.032	10
6	744	0.035	16
7	1972	0.13	12
8	4869	0.3	10
9	6398	0.37	20
10	37458	2.8	20

Table 2: Results of A* search with blind heuristic in the logistics domain

Task	**Expanded nodes**	**Time**	**Plan length**
1	11088	0,88	20
2	9612	0,78	19
3	4719	0,37	15
4	110109	11	27
5	24704	2,3	17
6	761	0,07	8
7	475503	64	25
8	26208	2,8	14
9	490850	57	25
10	429345	51	24

A.2 hADD heuristic

Table 3: Results of A* search with hADD heuristic in the blocks domain

Task	Expanded nodes	Time	Plan length
1	12	0.0092	6
2	14	0.0095	10
3	9	0.0075	6
4	40	0.06	12
5	19	0.023	10
6	47	0.074	16
7	143	0.34	18
8	45	0.1	10
9	105	0.23	22
10	47	0.12	22

Table 4: Results of A* search with hADD heuristic in the logistics domain

Task	Expanded nodes	Time	Plan length
1	21	0,047	20
2	39	0,069	21
3	16	0,027	15
4	38	0,089	30
5	133	0,29	17
6	9	0,014	8
7	75	0,23	28
8	15	0,037	14
9	32	0,08	27
10	25	0,092	24

A.3 hFF heuristic

Table 5: Results of A* search with hFF heuristic in the blocks domain

Task	Expanded nodes	Time	Plan length
1	11	0.0098	6
2	14	0.0092	10
3	8	0.0067	6
4	25	0.033	12
5	27	0.036	10
6	60	0.098	16
7	20	0.03	12
8	20	0.045	10
9	404	0.94	20
10	114	0.33	20

Table 6: Results of A* search with hFF heuristic in the logistics domain

Task	**Expanded nodes**	**Time**	**Plan length**
1	77	0,17	20
2	194	0,42	19
3	53	0,089	15
4	934	3	27
5	182	0,47	17
6	9	0,015	8
7	933	3,4	25
8	33	0,1	14
9	517	1,9	25
10	538	2	24

A.4 hMAX heuristic

Table 7: Results of A* search with hMAX heuristic in the blocks domain

Task	**Expanded nodes**	**Time**	**Plan length**
1	21	0.018	6
2	23	0.02	10
3	18	0.018	6
4	147	0.23	12
5	133	0.2	10
6	297	0.45	16
7	276	0.69	12
8	755	1.9	10
9	2556	5.6	20
10	5949	22	20

Table 8: Results of A* search with hMAX heuristic in the logistics domain

height **Task**	**Expanded nodes**	**Time**	**Plan length**
1	4886	11	20
2	4187	7,9	19
3	1206	2,2	15
4	74693	190	27
5	6206	14	17
6	281	0,45	8
7	202229	680	25
8	3606	9,9	14
9	200017	660	25
10	133521	460	24

B Tables of the enforced hill climbing search

B.1 blind heuristic

Table 9: Results of ehs search with blind heuristic in blocks domain

Task	Expanded nodes	Time	Plan length
1	100	0.0031	6
2	68	0.0038	10
3	66	0.0031	6
4	467	0.031	12
5	560	0.044	10
6	757	0.071	16
7	2166	0.4	12
8	4885	0.66	10
9	6496	1.8	20
10	36647	9.3	20

B.2 hADD heuristic

Table 10: Results of ehs search with hADD heuristic in blocks domain

Task	Expanded nodes	Time	Plan length
1	8	0.0043	6
2	69	0.043	24
3	10	0.0071	8
4	110	0.12	26
5	150	0.17	24
6	40	0.045	24
7	50	0.089	26
8	398	0.87	24
9	577	1.4	32
10	9194	37	48

B.3 hFF heuristic

Table 11: Results of ehs search with hFF heuristic in blocks domain

Task	Expanded nodes	Time	Plan length
1	58	0.042	14
2	13	0.011	10
3	15	0.0086	10
4	29	0.025	20
5	142	0.17	22
6	88	0.11	28
7	37	0.047	20
8	570	1.4	22
9	467	1.2	36
10	66	0.079	26

B.4 hMAX heuristic

Table 12: Results of ehs search with hMAX heurstic in blocks domain

Task	Expanded nodes	Time	Plan length
1	65	0.038	6
2	67	0.036	12
3	46	0.041	8
4	473	0.48	14
5	529	0.55	12
6	504	0.51	16
7	4825	9.5	18
8	4426	9.3	12
9	5586	5.1	20
10	50474	110	24

C Tables of the weighted A* search

Table 13: blind heuristic weighted A* logistics domain

Task	Expanded nodes	Time	Plan length
1	11088	0,88	20
2	9612	0,74	19
3	4728	0,36	15
4	110183	11	27
5	22654	2,1	17
6	902	0,0078	8
7	482536	59	25
8	29348	3,1	14
9	462795	55	25
10	421125	51	24

Table 14: hADD heuristic weighted A* logistics domain

Task	Expanded nodes	Time	Plan length
1	21	0,042	20
2	24	0,047	21
3	16	0,022	15
4	32	0,082	30
5	22	0,053	19
6	9	0,014	8
7	29	0,081	28
8	15	0,037	14
9	29	0,088	27
10	25	0,074	24

Table 15: hFF heuristic weighted A* logistics domain

height **Task**	**Expanded nodes**	**Time**	**Plan length**
1	22	0,043	20
2	22	0,047	19
3	18	0,032	15
4	31	0,095	27
5	21	0,052	17
6	9	0,016	8
7	29	0,09	25
8	15	0,038	14
9	28	0,1	25
10	28	0,09	24

Table 16: hMAX heuristic weighted A* logistics domain

Task	**Expanded nodes**	**Time**	**Plan length**
1	141	0,3	20
2	294	0,48	23
3	41	0,064	15
4	2457	6,2	27
5	159	0,27	19
6	64	0,128	8
7	9001	27	26
8	135	0,28	16
9	3669	11	28
10	667	1,8	28